

\$1

Washington Apple Pi



Volume 2

January 1980

Number 1

Highlights

Software Color Organ pg. 3

Review - APPLE-DOC pg. 2

Review - Tiny Pascal pg 3

In This Issue

	Page
EVENT QUEUE	1
CLASSIFIED ADS	1
EDITORIAL	1
SIG-NEWS	1
MINUTES	2
APPLE-DOC: A REVIEW - LEE HAUSMAN	2
SOFTWARE COLOR ORGAN - BRUCE F. FIELD	3
PASCAL (TINY PASCAL): A REVIEW - SANDY GREENFARB	3
ERRATA: GRAPHICS DRIVER FOR THE IDS 440 PRINTER - HERSCH PILLOFF	4
A PAGE FROM THE STACK - LIBRARIAN'S CORNER - DAVE MORGANSTEIN	5
STOCK MARKET DATA FROM THE SOURCE - HERSCH PILLOFF	5
MASTER CATALOG PROGRAM - HOWIE MITCHELL	7
CATALOG OF LIBRARY DISKS 1 - 6	8
AWARENESS TEST - HOWIE MITCHELL	9
A REACTION TO "IN MY OPINION" - CHARLES H. REINBRECHT	10
A QUICK AND DIRTY RAM TEST - JIM KELLY	10



COMPUTERS PLUS, INC.

6120 Franconia Road, Alexandria, Virginia 22310 703-971-1996

COMPUTERS PLUS, INC. carries a broad line of Micro Computers and peripherals as well as one of the largest book selections in the area.

Hours: 10:00 am - 9:00 pm Mon-Fri
10:00 am - 6:00 pm Saturday

Soon to be offering classes and seminars on all aspects of microcomputing.

Authorized dealers for: APPLE
Cromemco
Dynabyte
Micropolis
Northstar
soon to add: Thinker Toys

Authorized service: APPLE
Micropolis
Dynabyte



"The Plus Makes the Difference"

Officers & Staff

President	- John L. Moon	(202) 332-9102
Vice President	- Bernard Urban	(301) 229-3458
Treasurer	- Robert Peck	(301) 770-1954
Secretary	- Genevie Urban	(301) 229-3458
Members-at-Large	- Mark Crosby	(202) 488-1980
	- Sue Eickmeyer	(301) 953-7355
	- Sandy Greenfarb	(301) 674-5982
Editor	- Bernard Urban	(above)
Associate Editor		
Librarian	- David Morganstein	(301) 972-4263

Washington Apple Pi
PO Box 34511
Washington, D.C. 20034

EVENT QUEUE

Next meeting of Washington Apple Pi is Saturday, January 25, at 9:30 A.M. at George Washington University School of Engineering, Tompkins Hall (room 205), 23rd and H Streets, N.W. The topic is a panel discussion of available text and word processors.

NOVAPPLE will meet at 7:30 P.M. on January 24 at Computerland of Tysons Corner and on February 13 at Computers Plus in Franconia.

NYBBLES

You may have noticed strange spacings in some of the text in this issue. Genevie and I have begun to use our Come printer together with Easywriter to produce the newsletter, and haven't quite gotten on to the technique for using programmed proportional spacing. Right now, Easywriter is right justifying by inserting spaces between words. Maybe by next issue....

Members are reminded that 1980 dues are now payable. If you haven't renewed your membership please fill in the application form at the end of the newsletter and send it with your dues to Bob Peck.

The mailing label on the newsletter will tell you the status of your dues. The last four digits show the year and month your membership expires.

Classifieds

FOR SALE: APPLE carrying case, \$20 or best offer.
Chuck Reinbrecht, 299-5310.

Classified ads accepted from members 50 words or less at no charge provided the material is obviously non-commercial. Submit your classified 30 days in advance if possible, attention CLASSIFIED ADS, PO Box 34511, Washington, DC 20034.

EDITORIAL

by Bernie Urban

We begin with this issue our second year and, with your help, the newsletter will continue to improve. Many ideas have been voiced, tried and shelved, but it's time to try again. First, some good news and bad. Mark Crosby has switched jobs (and home phone numbers - it's 1980 you know), and that's good. The bad news is, because of the switch, he can no longer devote the considerable time and effort he has taken in the past to bring our newsletter to its current state. However, he promises to continue to contribute articles and such.

I have discussed the situation with several of you and I have come up with a partial answer. From now on you will see, on a monthly or bi-monthly basis, columns by several members on various topics. Dave Morganstein will cover matters concerning our software library and the disks of the month. Hal Weinstock will provide information on the APPLE information library he is setting up. Sandy Greenfarb promises to continue his column as time and inputs permit. I suggested to Phil Eastman (President of NOVAPPLE) that maybe someone from his group could pick up on an area needing attention - software for members without disk drives, e.g. a Tape Operating System (see Micro April, 1979) and other utilities. A column could be developed around that. I believe that someone should volunteer to write regularly or reprint materials of use to newcomers. Also, we need a treasurer - and whatever happened to our SIG's? Mark had suggested a column devoted to Dungeons and Dragons fans. Of course, we need and will continue to publish materials of appropriate nature that are contributed by anyone.

Regarding the mechanics of getting the newsletter out, I'm looking for volunteers with printer's ink in their blood. How about some art work? I've just (with Phil Eastman's help) gotten my Come up and running and Genevie and I will try for camera-ready copy on it from now on. \$ o o o - if you can provide your articles and materials on disk ready for printing, or in camera-ready form, we will be in business. Also, I will be asking for volunteers to carry copies in bulk to computer stores which they frequent. This will spread some of the workload. I'm open to other ideas.

SIG-NEWS

SOURCE INTEREST GROUP AND DIRECTORY

For Apple Pi members who are subscribers to the SOURCE, Chuck Reinbrecht has offered to create an Apple Pi SOURCE Users directory and act as interest group coordinator.

To participate, send 'MAIL' to Chuck at TCA257. Include how you would like to be listed in the directory. Any comments, requirements or suggestions regarding either the SOURCE or the Apple Pi SOURCE Interest Group are also solicited. Please include any additional user IDs, such as software suppliers, that you have found.

Chuck will integrate and assemble the input and send the results via 'MAIL' to those who participate. Further information and discussion will be supplied at the January meeting, including a report on replies received by then.

MINUTES

The Washington Apple Pi meeting of December 15 was called to order at 9:30 by Bernie Urban. He presented his idea that this meeting and future meetings should limit the business portion to 30 minutes, with the remainder of the meeting devoted to an organized instructive program, followed by the hands-on at the computers upstairs. Sue Eickmeyer agreed to be program chairman. The membership was asked for ideas on content of future programs.

Treasurer Bob Peck gave a brief financial overview of our club. He also read a letter to Washington Apple Pi from the George Washington University Medical Center stating their interest in a part-time programmer (10 hrs. per week) for the APPLE in Basic and machine language, at a reasonable negotiable salary.

The meeting was then turned over to Librarian Dave Morganstein who described the status of our program library and demonstrated on the APPLE some of the programs which are available for purchase. Sale of the disks after the meeting proved highly successful.

There were 57 attendees at this meeting.

NOVAPPLE Minutes for December 12, 1979

The meeting was opened by President Phil Eastman at 7:40 P.M. A brief business meeting was held due to the need to decide what to do about the INTERNATIONAL APPLE CORPS. The President explained that there is a \$50 initiation fee and there will be a need to pay dues for each member. A motion was made and passed that we pay the \$50 initiation fee and join the INTERNATIONAL APPLE CORPS.

Because of the vote the President asked the officers to re-examine the dues structure and report back to the membership by the second meeting in January.

Dr. Nick Cirillo proposed that we adopt a set of By-Laws. The President appointed a committee to look over the current By-Laws of Apple Pi and Call Apple and develop a set of By-Laws for NOVAPPLE. Dr. Nick Cirillo will chair that committee.

It was announced that all persons not paying their dues by December would be dropped from the Apple Pi mailing list due to the expense of carrying unpaid members. We will not be able to provide free copies to guests. Copies are on sale at various computer stores and can be purchased by interested individuals.

Mr. Kim Woodard presented more of his machine language tutorial. He finished the basic machine code instructions and is now ready to present a program on Sweet 16. This program will be presented on January 24, 1980.

NOVAPPLE Minutes for January 9, 1980

The meeting was opened by the President at 7:45 P.M. He stated that we do not as yet have our application for the INTERNATIONAL APPLE CORPS. He will try to get an application as soon as possible. The President appointed Mr. Theron Fuller as program chairman. He will organize our programs so more advanced notice can be given on presentations. Coming up in the near future are Sweet 16, a pilot language developed by Theron Fuller, Mountain Hardware demonstrations and an Applesoft tutorial. If you have a desire to present a program, please contact the program chairman or one of the officers of the club.

The program was presented by Mr. Gerald Eskelund on the subject of Low Resolution Graphics. It was a combination of a review of Nat Wadsworth's book "Introduction to Low Resolution Graphics" published by Scelbe Publication, a demonstration of the programs in the book, as well as an explanation of how you develop the programs. If you missed this program you may be able to see it again at an Apple Pi meeting in the near future.

The next meetings of NOVAPPLE will be January 24 at Computerland of Tysons and February 13 at Computers Plus in Franconia. The meetings will usually start around 7:30 P.M. so be there and help us all learn how to use the APPLE better.

Gerald Eskelund, Secretary

APPLE-DOC

A Review by Lee Hausman

APPLE-DOC is a handy package of utility programs to aid the Applesoft programmer in documenting his work. Consider the plight of long-winded programmers who, like myself, often fail to keep track of variables as they rattle along at the APPLE keyboard. Has the variable A1 been used yet? And if so, for what? Hmm... Or how about the poor Applelooney who discovers 10K into his program that he used Ctl-D for something other than DOS control and now he has to go through and make all those replacements (and hope he gets them all....). Well, these are exactly the kind of problems that APPLE-DOC handles with ease, via one of its three programs:

VARDOC - generates a complete variable table, showing each variable followed by each line number in which that variable appears. You may even enter "descriptors" which identify each variable. Output may be routed through video or a printer. Makes an excellent hardcopy record to go with your program listing.

LINEDOC - is quite similar to VARDOC and generates a cross-reference table for each line called by a GOTO, GOSUB, etc.

REPLACE - is the workhorse of the package, and allows you to locate, remove, or replace any variable, line number, or literal in a program! You can remove REM statements (great for compacting programs), change referenced line numbers throughout the program, or replace one variable with another, for instance. And any of these functions can be global or selective; leave the REM's at the top of your program containing your snazzy title and copyright info, and dump the rest.

Each program is loaded in above your existing program with an EXEC command, and takes over from there, occupying only 5.7K of memory. APPLE-DOC is available in disk or tape versions and comes profusely documented with 14 pages of instructions. Modifications for your printer are very simple and thoroughly explained. And the product has a Money-Back Guarantee!

I feel it is a nice addition to any programmer's library of utilities, and was my boss impressed when I presented my latest Aardvark Counting Program and the listing included those nice variable tables and line number cross-references! And the possibilities for cutting down on those programming headaches are limitless! APPLE-DOC is available from SOUTHWESTERN DATA SYSTEMS, P.O. Box 582, Santee, CA 92071 or possibly from your local retailer. Enjoy!

SOFTWARE COLOR ORGAN

by Bruce F. Field

I'm sure you have all seen the color organs that are on the market. They usually consist of a box containing different colored lights that pulsate and change color according to music that is played. This integer BASIC program for the APPLE is intended to approximate such a color organ. A low resolution graphics display with three colors (green, yellow, and red) is used with each of the colors representing the low, middle, and high frequencies of the music.

The reason for the title "Software Color Organ" is that no additional hardware, over and above your APPLE and cassette recorder, is required. The music is played on your recorder which should be connected to the cassette input jack on the APPLE. Pull out the plug that goes to the earphone jack on the recorder so you can hear the music, and then gently reinsert it so that it is making contact and sending the signal to the APPLE. Experiment with different volume control settings, although I think the program works best at moderate volume levels.

The program listing itself is undocumented to make it run faster, so here is a brief description of how it works. The program starting at line 400, clears the screen and prints the title. A machine language program is poked into memory by lines 420 to 480. The machine language program which is called at line 240 digitizes the music waveform from the cassette input port and stores the data in memory. Lines 250 to 270 perform the spectral analysis of the waveform, separating the low, middle, and high frequencies. A modified Fast Fourier Transform is used to do the analysis. All that is left now is to provide some kind of display. The subroutine at lines 20 to 90 erases and draws crosses on the low-res graphics screen. It first erases any cross that previously was drawn at location X,Y then it redraws a cross of color C and size S. Lines 120 to 200 repeatedly call the cross subroutine with different colors for the three different frequencies, with the size proportional to the magnitude of the frequency component. After updating the display the program continues to digitize more music.

For many reasons the program does not provide a true "picture" of the music. Most of these problems (the worst being quantization error, aliasing, and slowness of the integer BASIC) are insurmountable without additional hardware. However, the program does produce an interesting display that is fun to watch.

```
10 GOTO 400
20 COLOR=0: HLIN X-9,X+9 AT Y
30 VLIN Y-9,Y+9 AT X
40 COLOR=C
50 IF S<0 OR S>9 THEN RETURN
70 HLIN X-S,X+S AT Y
80 VLIN Y-S,Y+S AT X
90 RETURN
120 C=12:S=(64-T0-T1)/10
130 X=15:Y=19:GOSUB 20
140 X=24:Y=10:GOSUB 20
150 C=13:S=(T0-T1)/2
160 X=9:Y=9:GOSUB 20
170 X=20:Y=30:GOSUB 20
180 C=9:S=T2/2
190 X=9:Y=29:GOSUB 20
200 X=30:Y=20:GOSUB 20
240 CALL 4110
250 T0=PEEK(4098)+PEEK(4100)+PEEK(4102)+PEEK(4104)
260 T1=PEEK(4099)+PEEK(4101)+PEEK(4103)+PEEK(4097)
270 T2=PEEK(4104)-PEEK(4102)+PEEK(4100)-PEEK(4098)
280 GOTO 120
400 CALL -936
410 VTAB 23: TAB 11: PRINT "APPLE COLOR ORGAN"
420 POKE 4110,160: POKE 4111,8: POKE 4112,162: POKE 4113,8
430 POKE 4114,169: POKE 4115,0: POKE 4116,153: POKE 4117,0
440 POKE 4118,16: POKE 4119,173: POKE 4120,96: POKE 4121,192
450 POKE 4122,10: POKE 4123,169: POKE 4124,0: POKE 4125,121
460 POKE 4126,0: POKE 4127,16: POKE 4128,153: POKE 4129,0
470 POKE 4130,16: POKE 4131,202: POKE 4132,208: POKE 4133,241
480 POKE 4134,136: POKE 4135,208: POKE 4136,231: POKE 4137,96
490 GR
500 GOTO 240
600 REM SOFTWARE COLOR ORGAN
610 REM B.F. FIELD NOV 79
```

PASCAL (TINY PASCAL)

A Review by Sandy Greenfarb

32K, BISK: \$49.95
Programs International, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010

Tiny Pascal is a modified subset of the PASCAL programming language as defined by Kin-Man Chung and Herbert Yuen in the September thru November 1978 issues of BYTE magazine. It is an integer only language and, in fact, the documentation cites the analogy that Tiny Pascal is to a full Pascal what Integer BASIC is to Applesoft. Tiny Pascal is a software product and requires no additional hardware for its use. It is fully compatible with DOS and can run on a 32K machine, according to specifications. It is this reviewer's opinion, however, that practical use is not possible with less than 48K.

This review is oriented toward the product, not the language. Let it suffice to say that Pascal is a structured programming language that, in some ways, is similar to FORTRAN. The design of Pascal is such that it forces its users to adopt good programming practice and this is the probable reason for the current popularity of Pascal and similar languages.

The buyer of Tiny Pascal receives 27 pages of documentation and a diskette. I have had 20 years of computer experience but have never been personally exposed to Pascal other than by reading articles in magazines from time to time. I would say that it took me a good six hours or more of solid effort to feel that I was a "novice" Pascal programmer. I say this in order to justify my opinion of the documentation. The documentation is most definitely not tutorial. The buyer of Tiny Pascal, if not already familiar with the language, is advised to buy a "teachings" book on the subject. As reference, however, I find the documentation more than adequate, just a little too deep for beginners. By careful selection of his examples, the author augments the reference with a little advanced teachings, but this is only after one is familiar with the basic concepts.

The diskette contains several files. The major file, the system itself, contains a (Pascal system) monitor, editor, source program compiler, and an object program interpreter. This all occupies 0800 - 3fff in memory. 4000 - 4fff is the default for the Pascal variables stack, and the remainder of memory is used for compiling the program. In fairness to a previous statement, the owner of a 32K system has several means to increase source program size, i.e., decrease stack size, deactivate the editor and use its space for compiling, or deactivate DOS and manually save the (binary) file to cassette and write to diskette later on. More flexibility is still possible. The monitor and editor are both written in P-code (Pascal object code) and interpreted by the interpreter. In fact, their source files are provided on the diskette for

continued

experienced and/or brave users that wish to modify them. For the actual running of object programs, all that is really reserved for the system are parts of the zero page and \$0800 - 17FF (interpreter, reserved space for user functions/patches/enhancements, and monitor). The user can compile his programs in higher memory with a P-code origin at \$1800 (where the user intends to run it) and then move the compiled object program to \$1800. The end result is a self-contained Pascal entity which can be saved and run independent of the full system. One other significant option is that one program can address another independently produced (external) program, giving the capability to integrate larger efforts than are possible in one compile.

In view of the fact that the system partially overlaps the hires graphics area, Programma thoughtfully provided, on the diskette, a separate version of the interpreter which resides at \$4000 for use with page one hires graphics. The documentation states that Programma will be happy to assist you if you need it for other locations.

Tiny Pascal is primitive and machine independent in nature. In order to use the APPLE II to capability, a user would be forced to learn both the Pascal language and machine language. Tiny Pascal contains no integral commands for graphics, screen manipulation, advanced I/O, etc. Programma thought of this and provided two library files on the diskette containing Tiny Pascal procedures and functions to simulate these APPLE II capabilities and string handling. What they did not provide was a methodical way to integrate these libraries with user source programs. Again there exist "brute force" methods, as one can load a library and write a user program around it. To append one library to another (or, in general, to append two source programs), load the first, determine the high address used (stated by the system when the editor is accessed), go back to command level and RLOAD the second file at that high address. On this last subject I am vehement. Routine libraries are not an enhancement. They are a natural part of using a language such as this. I find fault with Programma in that I had to learn how to do it by my own experimentation and that neither this nor any other method was even mentioned in the documentation.

Two user programs are also on the diskette in source code, Cannibals and Detoken. Cannibals is a cute little game once you figure out how to play it. A couple words of documentation could have been provided. Detoken is a utility for creating a text file from an Integer BASIC program, but compiling it creates a "STACK FULL" error, forcing the user to modify the "Fence" before he/she has scarcely learned what it means. I personally consider that it would have been appropriate to contain one of the standard demonstrations/games with which APPLE owners are familiar. This would give them a basis for comparison and understanding. Nonetheless, the source programs do supplement the documentation. They do provide good examples for a beginning Pascal programmer and should be carefully scrutinized by each new user.

The documentation alludes to the fact that Tiny Pascal programs will run about twice as fast as Integer BASIC. I tested two simple programs, Rod's Color Patterns (from the APPLE II Reference Manual) and a short program to print the numbers from 1 to 5,000. In both cases, my Tiny Pascal version took longer. However, I find this inconclusive and tend to believe the documentation in general. Due to the different inner workings of both languages, I would expect that as size and complexity increased, so would apparent efficiency of Tiny Pascal over Integer BASIC.

In this review I have attempted to evaluate the product, giving both good and bad points of what the user might purchase and feel that, in fairness, a re-evaluation is necessary with one additional guideline, the price. The buyer receives more than the moneysworth. Some "moving paddle" games cost almost as much, and Tiny Pascal is not a game. It is a serious device for learning a language and, moreover, is a practical device for programming. If you are interested in the APPLE for more than just games, and if you are not considering the "high price spread" (the full \$500 hardware and software version), then I strongly recommend that you consider this product.

ERRATA: GRAPHICS DRIVER FOR THE IDS 440 PRINTER

by Hersch Pilloff

The HIRES DUMP listing printed in the Washington Apple Pi Newsletter of December 1979 (pg.23) contains an error in LINE 210. This error causes several spurious dots to be printed in the extreme upper left hand corner of the graph. Bob Bolster correlated the dot pattern with a carriage return and a line feed and suggested that the colon following the last print statement be changed to a semi-colon. This corrects the problem provided that all REM statements are removed from this line. The REM statements will be placed in LINE 209. The essential point to be considered here is that when PRINTER is run with CALL 768, the number of characters per inch (cpi) or equivalently the horizontal dot spacing (hds) is determined by the default settings of the DIP switch on the IDS 440. If a different hds is desired, the CALL 768 must be followed with a CALL to the appropriate hds. In the event that this second CALL would correspond to the default hds, this routine won't work. Because the default setting of the IDS 440 was 12 cpi, CALL 940 should not be used here. The factory warning against using 16.5 cpi (CALL 945) for plotting still applies.

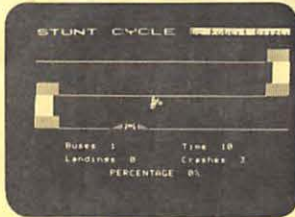
```
209 REM LINE 210//REM CALL PRINTER // OPTION
    INSERT 'CALL <> 940' PRINT SIZE // SET
    GRAPHICS
```

```
210 Call 768: CALL 935: PRINT: PRINT C$;
```

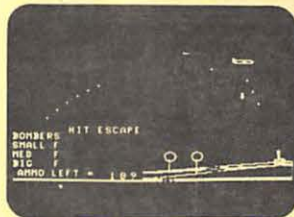
```
*****
* At the time this newsletter goes to press *
* Washington Apple Pi has 119 paid members. *
* Of these, all but 4 members are within *
* "easy commuting distance" of our meetings. *
*****
```

NEW APPLE II SOFTWARE

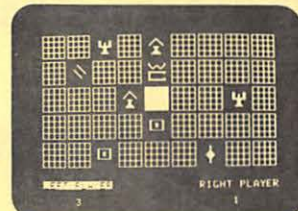
PROGRAMMA Software Program Products



STUNT CYCLE \$15.95



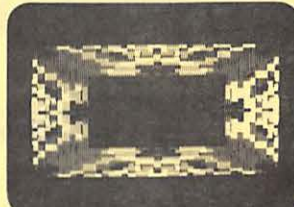
BLITZKRIEG \$15.95



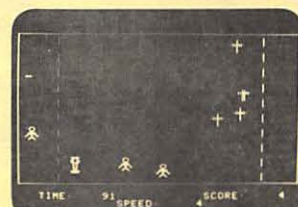
FRUSTRATION \$ 9.95



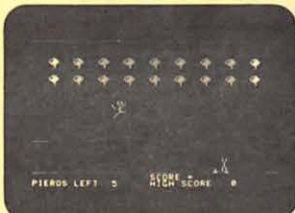
PERPETUAL CALENDAR \$ 9.95



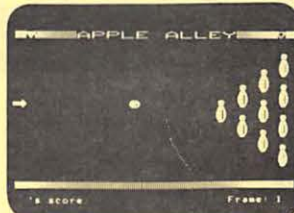
LORES HYPERPAK \$ 6.95



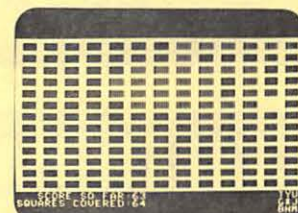
DEATH RACE \$15.95



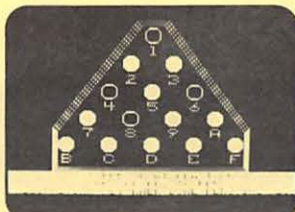
CLOWNS & BALLOONS \$15.95



APPLE ALLEY \$ 6.95



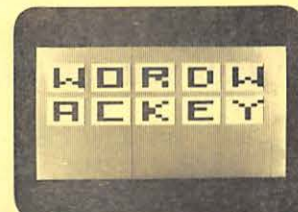
MOUSE HOLE \$ 6.95



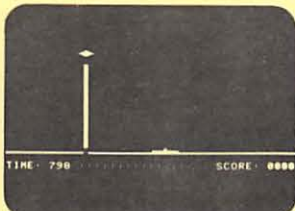
PEG JUMP \$ 9.95



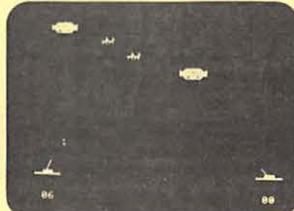
I-CHING \$15.95



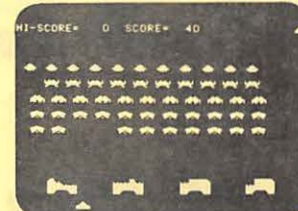
WORDWACKY \$ 6.95



ALIEN INVASION \$ 9.95



GUIDED MISSILE \$15.95



APPLE INVADER \$15.95

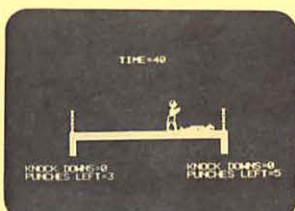


BASEBALL \$15.95

All orders must include 3% postage and handling with a minimum of \$1.00. California residents include 6% sales tax.
VISA MASTERCHARGE



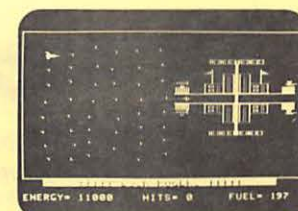
BREAKTHRU \$ 9.95



BOXING \$ 9.95

PROGRAMMA INTERNATIONAL, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579
384-1116
384-1117

Dealer Inquiries Invited



STAR VOYAGER \$15.95

ComputerLand[®] and apple II

For the best in personal computing

SOFTAPE ™

Personal Software™

D. C. Hayes Associates, Inc.

MICROCOMPUTER PRODUCTS

CENTRONICS®

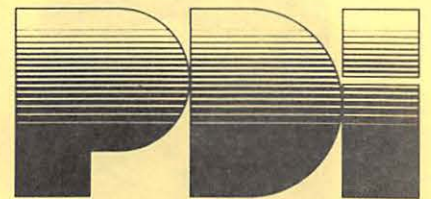
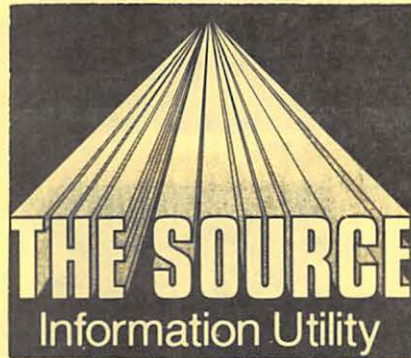


Mountain Hardware, Inc.



Integral Data Systems, Inc.

MUSE



**houston
instrument**

Heuristics
INC.



Automated Simulations



ComputerLand®

We Know Small Computers.

ComputerLand/Tysons Corner

8411 Old Courthouse Road at Rt. 123 — 893-0424



A PAGE FROM THE STACK

- Librarian's Corner-Dave

Before I even start these notes...HELP! If everybody out there contributes one program, either of your own or typed from a magazine, we'll never have a shortage of new material. Outside of three or four big contributors, very few members have added to the library. Come on now, we need your help.

Well, right now we have eight volumes in the Library. Vols. 1, 2 and 3 are utility disks, while Vols. 3 - 7 are games. In the next month a scientific / math disk and an educational disk should bring us up to a total of ten.

Utilities seem to be of keen interest. However, they are frequently very short programs and it takes a lot to fill a disk. (Besides which, you members are not submitting many. Ahem.)

Each month I will try to review a few disks. Hopefully, we will generate new ones faster than they can be reviewed. I can dream, can't I? This month I'll describe Volumes 1, 3 and 4.

VOLUME 1 : UTILITIES

IMPROVED CATALOG - Gives a catalog showing free space and the starting address and length of binary files. (In your choice of hex or decimal!)

DISK AIDE - An excellent package for examining any track/sector on a disk. Also allows you to modify and save back to disk. I have recovered a crashed catalog with this kind of program.

DISK MAP - A colorful graphic display of tracks and sectors, indicating used, unused and bad sectors.

TONY'S SUBROUTINE PAK - An integer package for renumbering programs, appending programs, doing hex-decimal conversions, setting scrolling windows, flipping between display pages, and other goodies.

LOCK DISK - Lock all programs on a disk.

PROGRAM ELIMINATOR - Delete unlocked programs.

SUPERCATALOG - Search catalog for selected character strings.

DISC SPEED - Check and adjust your drive's speed.

LOOP - Loop through a "catalog" on many disks.

(Great when used with a printer!)

MEM TEST - Like the name implies.

CAT. TO MENU - A great HELLO program. Presents a catalog and allows program selection by printing one letter.

SYMBOL TABLE XREF - Prints every variable in your program and all lines in which they are referred to.

LINE # XREF - As above with line number references (JOSUB, GOTO, and THENs.)

STOP LIST - Cheaper than an auto start ROM!

SPLIT CATALOG - More catalog on your screen.

FREE SECTORS - Tells you how much disk space remains.

OKIDATA.OBJ - A printer driver for an OKIDATA 110 (Guess who owns one!)

B/BSTAT - Starting address and length of binary files.

VOLUME 3 : GAMES

(Assembled by Sandy Greenfarb)

Since these are mostly old favorites I won't go into heavy detail.

TOWERS OF HANOI - The buddhists say that if you move all 64 disks the world will end ... be careful!

TENNIS - Remember the original PONG?

ROULETTE - Bet you can't play only once.

MIDWAY - Dive bomb a carrier while avoiding the ack-ack.

COLOR TEXT - A little graphics.

MERRY XMAS - Seasons greetings.

BAGELS - An old favorite.

DRIP - Visuals with sound, get a plumber.

DIGITAL CLOCK - You don't have to wind it.

POKER - It bluffs.

KENO - I don't know how to play.

NITEMARE # 6 - The object of the game is to figure out the object of the game.

SLOT MACHINE # 2 - Finance new APPLE peripherals.

MASTERMIND - A text version with numbers.

POET - Not my cup of iambic pentameter.

... and many others.

VOLUME 4 : GAMES

These games are longer and more complicated than Volume 3.

PRO FOOTBALL - See if you can make it to the Super Bowl.

ATOM 20 - From Creative Computing, it's the end of civilization. Can you survive in the "adventure" game?

CIVIL WAR - I'll "Grant" you this may not end too happy "Lee".

LEM - Land your lunar module.

DR. Z AND ELIZA - A cheaper way to have your psychological ruffles soothed. Makes your APPLE seem to be a person. (What do you mean, it isn't?)

RED GNIK - A Hammurabi-like game.

DEEPSPACE - Also from Creative Computing. Pick your ship and go after the enemy. Not graphics.

ADVENTURE - Not the original but a good in-RAM game so it moves along more quickly.

QUEST - From Byte, another adventure-like game.

(Typed in by our able President.)

STOCK MARKET - Another way to finance further APPLE purchases. (If you win, of course...)

STARSHIP ATTACK - From Kilobaud, a lo-res action game. Shoot the enemy before he destroys you or all your grain ships.

STOCK MARKET DATA FROM THE SOURCE - Hersch Pilloff

Investment technical analysis can be greatly facilitated using microcomputers. The APPLE II is well suited for this type of application because it provides a powerful floating point basic which can handle large multidimensional arrays and can easily generate high resolution graphics.

The human problems associated with manually inputting (without errors!) even modest amounts of daily stock data from the newspaper usually prompts consideration of alternative means of data acquisition. For the small investor the optimum approach is to subscribe to an electronic financial data retrieval service and download the desired data into the APPLE's RAM for storage in a disk text file for later analysis. Since the charges for this service are determined in large measure or totally by the connect time, this procedure minimizes the user's daily charge. Unfortunately, the monthly charges for daily access on these systems, which are aimed at the serious professional investor, are too high (estimated \$60 to \$125 per month minimum) for most small investors. Thus the introduction during the summer of 1979 of a new time share system, the Source, which promised stock market data and a non-prime time (other than 7 AM to 5 PM E.S.T. Monday thru Friday) connect charge of \$2.75 per hour, offered for the first time the possibility of a relatively inexpensive electronic financial data base. Regrettably, the current implementation of the Source is unsatisfactory in several important respects.

continued

This article was written in the hope that the deficiencies to be described below will be rectified and the Source stock market data will then provide a valuable service. The remainder of this article will describe some of the author's experiences in using the Source to obtain stock market data, commercially available software for downloading to the APPLE, and subsequently writing a disk text file.

At the present time the Source financial data bases are contained in UPI UNISTOX and a specific data base is associated with a given report number. For example, UNISTOX report 139 provides the current closing prices on the NYSE, report 149 closing prices on the AMEX, and report 105 closing prices OTC. The remainder of this article, insofar as it is report specific, will be limited to report 139.

Report 139 contains the UPI wide-line stock report. It should be noted that the UPI abbreviated stock names are in many cases different from those used by other wire services, e.g., Dow-Jones or the ticker tape. Moreover, because the total number of spaces allocated to the abbreviated name and current dividend is fixed, a change in the dividend requiring more line spaces may result in the abbreviation being shortened.

The most convenient approach for downloading data from the Source to the APPLE's RAM is to purchase (\$25) the "APPLE II Terminal Program, Source Version" written by Craig Vaughn and marketed by Peripherals Unlimited, Inc. This program provides, in menu selectable form, a number of terminal-type utility programs including one which downloads user files from the Source into the APPLE and then writes an APPLE disk text file. The user must first create and store a file on the Source. Then, using the Source line editor, both the beginning and end of that portion of the file to be downloaded must be marked by insertion of a Ctl-W. Because the commands used on the Source to create this file also write instructions, and comments as well as data, this is an easy way to eliminate the non-data portion of the downloaded material. After this file has been downloaded, it should be deleted from the Source in order to avoid additional disk storage charges. These terminal programs are coded in machine language, provide real 300 Baud response, and are fully compatible with both the APPLE Comm Card and an acoustic coupler or the D.C. Hayes Micromodem.

The remaining task is to write a file on the Source which contains the desired data. This is accomplished by permanently storing a very simple driving routine on the Source which when executed will generate a data file. This resident driver uses two recently added Source commands, CO and COMO. Very briefly, the CO command permits using a file containing a series of commands which are sequentially executed by typing CO <file name>, whereas the COMO command writes all subsequent interaction between the user's terminal and the Source into a file stored in the user's disk space. The following example shows a typical driver file, NYSE, which when executed by typing "CO NYSE", will cause a file, DATA, to be stored on the Source.

```
COMO DATA -N
UNISTOX
139
C B S
EAST KO
GOODYR
VIACOM
(This line must be left blank!!!)
QUIT
COMO -E -T
TY DATA
```

The last line TY DATA is optional and will list the file data on the video monitor. It is also helpful for indicating that the data search has been completed because it is much easier to see a full screen listing than the return of the Source cursor.

As noted before, the file DATA contains, in addition to the data on the 5 stocks which were listed in the driver, superfluous text including prompting instructions, etc. (In fact, for reasons to be discussed, more than 5 stocks can appear in these data.) The actual stock data can be separated from the remainder of the file by insertion of Ctl-W at the beginning and end of the data portion so that only the essential information is downloaded. The Ctl-W will appear as ASCII 227 (octal) when it is sent back from the Source.

When this text is downloaded to the APPLE, it is likely to include some lower case letters. For example, OCCDPET is actually downloaded as JccdpEt and lower case letters are sometimes used to provide additional information when they are prefixed to numerical values. In order to read this on the APPLE's monitor and to avoid potential problems in writing or calling disk files identified by stock names, it is desirable to convert all lower case characters to upper case. In addition, the data string for each stock contains information that usually would be stored on a daily basis, such as yearly high, yearly low, etc., and this information should be stripped from the string before each individual stock file is eventually appended.

While the above procedure works very well, the overall value of this system is rendered nearly worthless because of serious deficiencies with the Source. The minor problems begin with signing on during non-prime time. Typical sign-on times of 2 to 5 minutes are common and the SYSTAT command often indicates that there are 40 plus users on the Source. Running CO NYSE usually requires 20 to 40 minutes to search for 10 stocks. The search algorithm is very, very slow and would appear to compare every stock name in its entirety before executing the logic statements. For example, the time required to search for a single stock name beginning with an A is very much less than that required for a name beginning with a W. Moreover, the search routine only considers a limited number of characters in the string. The consequence is that OCCDPET in the NYSE driver not only retrieves OCCDPET but also OCCDPETPF. Even entering OCCDPET in NYSE with two trailing blank spaces doesn't change this result. In this way more output than requested can appear. These extra entries must then be deleted using the Source line editor prior to downloading. Finally, once the search routine has begun it can only be terminated by breaking the phone connection to the Source. Meanwhile the file DATA will store all information up to this time.

The most serious problems I have experienced involve missing data. First, there have been several occasions when an entire day's transactions have never appeared in UNISTOX. Second, and quite common, is the observation that, in a search of only 10 stocks, some data is missing. In this case a particular stock will be totally missing from UNISTOX (report 139). The missing stocks seem to vary without any discernable pattern. This intolerable situation has been repeatedly brought to the attention of the Source through their Customer Service representatives.

(Editor's note: The opinions expressed in the above article are those of the author and do not necessarily represent those of Washington Apple Pl.)

MASTER CATALOG PROGRAM

by Howie Mitchell

The following is a master catalog program which works well for finding diskettes containing specific programs.

```

0 REM >>> AUTO-SAVER AT L88 <<<
1 REM *****
*
* HOWIE MITCHELL
* 7823 SW. 55TH PL.
* GAINESVILLE, FL. 32601
* OCTOBER, 1979
*
*****
9 TEXT : PRINT CHR$(4);"MON O,
I,C": GOTO 25550
10 HOME : PRINT "*** LIST OF PRO
GRAMS & VOL. NOS. ***": PRINT
15 DASH$ = "-----"
20 PRINT " TYPE IN THE FIRST LE
TTER(S) OF THE PROGRAM TI
TLE YOU WANT, AND I WILL TEL
L YOU WHERE IT MAY BE FOUND.
": PRINT
25 PRINT " (OR: TYPE IN THE ";
INVERSE : PRINT " VOLUME NU
MBER ": NORMAL : PRINT " AN
D I'LL LIST ITS CONTENTS !)"
: PRINT
30 PRINT " (JUST PRESS 'RETURN'
IF YOU WISH TO SEE THE EN
TIRE LIST.)"
40 PRINT : INPUT " ?? ";BEG$
41 IF BEG$ = "" THEN BEG$ = " "
45 PRINT
50 READ PGM$,VOL
52 IF PGM$ = "END" THEN PRINT :
PRINT " (END OF LISTING.)":
GOSUB 26000: END
53 IF PEEK (37) > 19 THEN GOSUB
85
55 IF BEG$ = " " OR LEFT$(PGM$,
LEN(BEG$)) = BEG$ OR VAL
(BEG$) = VOL THEN GOSUB 80:
PRINT PGM$
65 IF PEEK (37) > 20 THEN GOSUB
100
70 GOTO 50
80 PGM$ = PGM$ + LEFT$(DASH$,33
- LEN(PGM$)) + " " + STR$(
VOL): RETURN
85 PRINT : INPUT "(PRESS 'RETURN'
/ TO CONTINUE; E TO END.)";H
OLD$: IF HOLD$ = "E" THEN UTAB
23: END
86 HOME : RETURN
88 REM *** UPDATE SAVER ***
90 D$ = CHR$(4):NAME$ = "MASTER
CATALOG"
94 PRINT D$"SAVE ";NAME$
96 END
99 REM *** DATA: ***
LINE #100-25500.
25501 DATA END,0
25550 REM *** INTRODUCTION ***
25555 HOME : HTAB 7: PRINT "***
MASTER CATALOG ***": HTAB 7:
PRINT "-----"
25560 PRINT : PRINT " THIS PROG
RAM CAN BE USED TO COLLECT
ENTRIES FROM ALL YOUR DISKE
TTE CATALOGS,THUS DEVELOPING
A 'MASTER CATALOG'.": PRINT
25565 PRINT " YOU WILL THEN BE
ABLE TO LOCATE WHICH VOLUME
NUMBER CONTAINS ANY DESIRED
PRO-GRAM, AS WELL AS LISTIN
G THE CONTENTS OF ANY GIVEN V
OLUME.": PRINT

```

```

25570 PRINT " WHEN YOU HAVE YOU
R CATALOG UPDATED, A 'RUN 88
' WILL COPY THE CATALOG PROG
RAM ONTO YOUR DISKETTE.": PRINT
26000 REM *** CATALOG UPDATE ***
26002 PRINT
26005 PRINT "(E=END: R=RUN: U=UP
DATE MASTER CATALOG)": PRINT
"*** WHICH? *** ": GET WHUT
$
26010 IF WHUT$ = "R" OR ASC (WH
UT$) = 13 THEN PRINT "*** R
UN ***": FOR T = 0 TO 1000: NEXT
: RUN 10
26020 IF WHUT$ = "E" OR WHUT$ <
> "R" AND WHUT$ < > "U" THEN
PRINT "*** END ***": END
26030 PRINT "*** UPDATE ***"
26035 FOR T = 0 TO 1000: NEXT
26040 TEXT : PRINT CHR$(4)"MON
O,I,C": HOME : PRINT "*** M
ASTER CATALOG UPDATE ROUTINE
***"
26045 HTAB 5: PRINT "-----"
26050 PRINT : PRINT " THIS ROUT
INE WILL UPDATE YOUR MASTER
CATALOG DATA....."
26055 PRINT : PRINT " LOAD YOUR
DISK UNIT WITH ANY DISK THA
THAS CATALOG DATA YOU WISH T
O ADD.": PRINT : PRINT " TH
EN: PRESS THE 'RETURN' KEY."
26057 PRINT
26060 PRINT " IN A MOMENT, YOU
WILL BE ABLE TO COPY THE CAT
ALOG IN THAT VOLUME (BY USIN
G THERIGHT-HAND ARROW ( '>>
' ) + 'REPEAT'. "
26062 PRINT
26065 PRINT " FOR MAXIMUM ELEGA
NCE, PLEASE DO NOT INCLUDE
ANY SPACES "; INVERSE : PRINT
"TO THE RIGHT"; NORMAL : PRINT
" OF THE PROGRAM NAMES !": PRINT
26070 PRINT : PRINT : INPUT "***
PRESS 'RETURN' TO CONTINUE.
***";HOLD$
26075 HOME : PRINT CHR$(4)"CAT
ALOG": PRINT " PRESS 'RETUR
N' TO LOWER CURSOR. USE '>>
->' ARROW (+ 'REPEAT') TO COPY
THE NAME,OR JUST TYPE IT. E
NTER '*' WHEN DONE."
26080 LEVEL = 1: REM : THIS IS
USED FOR PLACING THE CURSOR
INITIALLY AT THE TOP, FOR
COPYING CATALOG NAMES.
26085 DIM NAME$(30): PRINT : FOR
P = 1 TO 30
26090 UTAB 23: PRINT "
": UTAB 23: INVERSE
26095 PRINT "PROGRAM #";P;" NAME
?"; HTAB 6: UTAB LEVEL: INPUT
NAME$(P):LEVEL = PEEK (37) +
1: IF NAME$(P) = "" THEN P = P
- 1
26100 IF NAME$(P) = "*" THEN NORMAL
: GOTO 26200
26105 NORMAL : NEXT
26200 REM
*****
*
* MAKE CATALOG FILE
*
*****
26205 HOME : PRINT "*** THE PROG
RAM NAMES I HAVE ARE: ***": PRINT
: FOR N = 1 TO P - 1: PRINT
"NAME #";N;": ";NAME$(N): NEXT

```

```

26210 PRINT : INPUT " PLEASE TY
PE IN THE NUMBER OF THIS
CATALOG: ";C
26212 IF C < 1 OR C > 254 THEN PRINT
: PRINT CHR$(7)"SORRY, THA
T IS OUT OF RANGE OF POSSIBL
E VOLUME NUMBERS.": GOTO 262
10
26215 PRINT CHR$(4)"MON O,I,C"
: FILNAM$ = "CATALOG FILE": D$
= CHR$(4)
26220 PRINT D$"OPEN ";FI$
26225 PRINT D$"DELETE ";FI$
26230 PRINT D$"OPEN ";FI$
26235 PRINT D$"WRITE ";FI$
26237 PRINT "DEL ";C * 100;",";C
* 100 + 100
26240 FOR N = 1 TO P - 1
26245 PRINT C * 100 + N"DATA"NA$(
N);": NEXT
26247 PRINT "DELETE ";FI$
26250 PRINT D$"CLOSE ";FI$
26255 REM
*****
*
* "EXEC RETRIEVAL"
*
*****
26260 PRINT D$"EXEC ";FI$
26265 END

```

The following receipt was brought to our attention by Bruce Field. It originally appeared in IBM-USER, a national bureau of Standards Newsletter, Vol. 1, Issue 5, September. The benchmark programs referred to are the ones published in the October 1977 issue of Byte.

LETTERS TO THE EDITOR

In issue #2 we published a list of seven popular benchmarks for testing the speed of Basic interpreters on small systems. This month's letter from Seymour Haber refers to these tests.

Subject: Microcomputer Benchmarks

The times for your benchmarks 1-7, when done in Applesoft on my Apple II, are:

Benchmark no.	Time/seconds
1	1.4
2	8.5
3	16
4	18
5	19.5
6	28
7	45

I'd like to add some comments on the benchmarks and on timing them:

continued

- I don't think the timing on Bench #1 is meaningful. The reason is that all these timings include 2 reaction times which will vary from one experimenter to another. The timing of the start of the run, in particular, will vary --one experimenter will start his stopwatch as he presses the "return" after typing "run", while another will start the stopwatch only after seeing the word "Start" appear on the screen! All in all, I'd expect a systematic variation of as much as 1/3 second on each benchmark, as one varies experimenters.
- For the above reasons, I think the reported bench times should be rounded to the nearest second, not tenth of second.
- I thought the PET does 10 sig. fig. arithmetic and shows 9.
- I'd like to see the benchmarks done on an H-P 9845. I think it's quite fast. Also on the 1108?
- I'd like to propose a few other benchmarks, as listed below. Numbers 1a and 2a are intended as substitutes for nos. 1 and 2, to overcome the timing problems. The Apple's timing on them is 33 seconds and 42 seconds respectively. Numbers 8 and 9 are intended to reveal some peculiarities in the

handling of numbers by some BASICs. Note the different lengths of the loops --the Apple took 14 1/2 seconds for number 8, but 81 seconds for number 9!

#1A:

```
300 PRINT "START"
400 FOR K = 1 TO 25000
500 NEXT K
700 PRINT "END"
800 END
```

#2A:

```
300 PRINT "START"
400 K = 0
500 K = K + 1
600 IF K < 5000 THEN 500
700 PRINT "END"
800 END
```

#8:

```
300 PRINT "START"
400 FOR K = 1 TO 1000
450 A = 1E30
500 NEXT K
700 PRINT "END"
800 END
```

#9:

```
300 PRINT "START"
400 FOR K = 1 TO 1000
450 A = 1E-30
500 NEXT K
700 PRINT "END"
800 END
```

I don't know for sure how many precision digits Apple-soft works with. All I know is that the numerical range is roughly $+1E+38$ and that each number occupies 5 bytes. This suggests 9 1/2 significant figures to me.

Seymour Haber, 711

Following is a catalog list of Library Disks 1 - 9. We will try to publish the remaining disks next month.

```
001 I HELLO
001 I IMPROVED CATALOG
001 I DISK AIDE BOC.
001 I DISK AIDE MAC.
001 B DISK MAP
001 I TONY'S SUBROUTINE PAK
001 I LOCK DISK
001 I DISK PROGRAM ELIMINATOR
001 I SUPERCATALOG.BOC
001 B SUPERCATALOG.O
001 I DISC SPEED INFO
001 I DISC SPEED TEST
001 B DSPEED.OBJ
001 I L O O P
001 I SLOW MEM TEST
001 I FAST MEM TEST
001 I MEMORY SPY
001 I CAT. TO MENU
001 B SYMBOL TABLE XREF
001 B LINE # XREF
001 I STOP LIST DOC
001 B STOP LIST
001 I SPLIT CATALOG
001 I FREE SECTORS
001 B ONIDATA.OBJ
001 I B/BSTAT
```

```
002 - FREE SECTORS = 20 (128K)
002 I HELLO
002 I SWEET 16 DISASSEMBLER
002 I DISK TRANSFER
002 I DEBUGGING AID
002 B B.FIND
002 I FIND TEXT TONER
002 B B.TRACE
002 I PROGRAM TRACE
002 A MASTER CATALOG 10K
002 B B.MAS.CAT.10K
002 I TED II SAVE/RETRIEVE
002 I TED START
002 I TED3
002 B YLD.DISK
002 I BASIC FILE PRINTER
002 A DOS UTILITY #1
002 I CODES FOR 2.02
002 I PASSWORD MGR
003 - FREE SECTORS = 1 (128K)
003 I INTRO
003 I TOWERS OF HANDEL
003 I TENNIS
003 I ROULETTE
003 I MIDWAY
003 I COLOR TEXT
003 I MERRY CHRISTMAS
003 I BAGELS
003 I BRIDGE
003 I BILLIARD 1.00K
003 I POKER
003 I KENO
003 I SEA CHASE
003 I NIGHTMARE II
003 I SLOT MACHINE II
003 I COLOR WOND
003 I XMAS TREE
003 I MASTEMIND
003 I KALEIDOSCOPE
003 I MOUSE MAT
003 I BANANA BELL
003 I SAUCER WAR
003 I THE WAZZ
003 I BLACK JAG
003 I SHOOTING STARS II
003 I A TRILLION STORIES
003 I POET
003 I GARYS GUNNY
003 I COLOR LIFE
003 I AWAY
003 I SEVENS
003 I OTHERS
003 I MAD DOG
003 I BIRTHDAY
003 I BANANAS
003 I ANDROMEDA SWIM
003 I CRAZY
003 I SQUARES
004 - FREE SECTORS = 37 (128K)
004 I HELLO
004 A PRO FOOTBALL
004 I ATOM 20
004 A CIVIL WAR
004 A LEM
004 I DR. Z
004 I RED SNIP
004 A DEEPSPACE
004 A ADVENTURE
004 A ELIZA
004 I QUEST
004 A STOCK MARKET
004 I STARSHIP ATTACK
005 - FREE SECTORS = 10 (128K)
005 I HELLO
005 I BLACK BOX
005 I SHOOT OUT
005 A HUNT THE WUMOUS
005 I BEGINNER MATH
005 I SPELLING BEE
005 I APPLESTAND
005 I LEGACY/TANK II
005 I LD RES FOOTBALL
005 I TOGNAZZINI GUNSS
005 I NIGHTMARE SWIMPAK
005 I SIMPLE SIMON
005 I BRAIN BUSTERS
005 I CRYPTOGRAM
006 - FREE SECTORS = 40 (10K)
006 I HELLO
006 I STARWARS II
006 I DEATH STAR
006 B SPACE ADVENTURE
006 I HI-RES BREAKOUT
006 B HI RES BREAKOUT.HC
006 I INTERACTIVE BASEBALL
006 A INSPECTOR CLEW.SU
006 A ROLE-PLAYING STARWARS
006 A TREK 79
006 I HAUNTED HOUSE
```

AWARENESS TEST

by Howie Mitchell

The following program shows a curious phenomenon of perception.

```

4 TEXT
5 HOME: VTAB 2: HTAB 3: PRINT "***HOW MANY
  P'S DO YOU COUNT?"
6 HTAB 7: PRINT "-----"
7 VTAB 6
10 L$ = "*****"
12 FILL$ = ""
15 GOSUB 100
20 L$ = " FINISHED FILES ARE THE RE- *"
25 GOSUB 100
30 L$ = " SULT OF YEARS OF SCIENTIF- *"
35 GOSUB 100
40 L$ = " IC STUDY COMBINED WITH THE *"
45 GOSUB 100
50 L$ = " EXPERIENCE OF MANY YEARS. *"
55 GOSUB 100
57 HTAB 5
60 L$ = "*****"
65 PRINT L$: POKE 34, PEEK (37) + 1
66 PRINT : PRINT
70 INPUT " WHAT'S YOUR GUESS ? ";GUESS
71 TRY = TRY + 1
72 IF GUESS = 6 THEN 77
75 IF GUESS < > 6 AND TRY < 5 THEN PRINT
  " NOPE. TRY AGAIN." : GOTO 70
77 IF GUESS = 6 THEN PRINT : PRINT
  CHR$(7);" YOU GOT IT !!" :
  TEXT : VTAB 23 : END
80 GOTO 200
90 TEXT : END
100 HTAB 5 : FOR N = 1 TO LEN(L$)
105 IF ANS$ = "Y" AND MID$(L$,N,1) = "P"
  THEN INVERSE : PRINT CHR$(7);
110 PRINT MID$(L$,N,1); NORMAL : NEXT :
  PRINT : HTAB 5 : PRINT FILL$ : RETURN
200 HOME : PRINT : INPUT " GIVE UP ? ";ANS$
205 IF LEFT$(ANS$,1) = "N" THEN TRY = 0 :
  GOTO 70
210 IF LEFT$(ANS$,1) = "Y" THEN TEXT :
  HOME : GOTO 7
215 IF ANS$ < > "Y" OR ANS$ < > "N" THEN
  PRINT "SORRY. I MISSED THAT." : FOR
  T = 0 TO 1000 : NEXT : GOTO 200
300 REM *****
  *
  * FROM: BILL PAINE *
  * SILVA MIND CONTROL *
  * P.O.BOX 14075 *
  * GAINESVILLE, FLA. *
  * 32604 *
  * *****
305 REM *****
  *
  * PROGRAM: HOWIE MITCHELL *
  * 7023 SW. 55 PL. *
  * GAINESVILLE, FL. *
  * 32001 *
  * (NOV. 13, 1979) *
  * *****

```

π

A REACTION TO "IN MY OPINION"

by Charles H. Reinbrecht

"I am writing this to you as a result of reading the In My Opinion column in the Washington Apple Pi newsletter. Not in reply, but rather additive and free form as your well done article triggered some ideas and needs.

First, some perspective. Got my APPLE in June; got started on the Source in September; went to one 'Pi' meeting in Sept. or Oct. (included Inter. Apple Corps plans). Am a problem solver, not a systems person; interested in answers not experimenting, programs that work not half done, organizations that are organized and helpful, not a group of technical experts who like to talk in free form mode. With that background let me comment on your article:

1. I came looking for, not to give. But am interested in a fair exchange. One item that would help is if 'Pi':
 - a. Had a place on the membership / renewal form for names of purchased programs & evaluation.
 - b. Use this, plus other sources, to add a software evaluation col. to the newsletter, including copies of MICRO's column.

I realize from your points that I am not too new to give and hereby offer to provide a 1/2 to 1 hr. discussion on the tax impacts with alternatives of owning an APPLE. (I work for IBM, but work & teach for IBM Block part-time and just registered a tradename in Montgomery County.)

2. The advent of the library is great and the price is certainly right, but isn't there a catalog of what is on each disk? If needed, I'll offer to put one together, if I can borrow the disks. (I will buy what I need but want to know what I am getting.)

3. Meeting agendas in an organized fashion are needed. Also, it would be helpful to know who is who at the meeting. Both for visitors and for new members this would help, but I am sure there are older members who don't always know who is in the front of the room or talking on the side.

4. Special Interest Groups would be very useful as would area groups. Geographic area, that is, for getting together between meetings. My interests are portfolio tracking, income taxes, business simulations and business planning.

5. Don't really know or understand, the 'share' in the club bit. Might be a good idea, or maybe we should increase disk and/or dues. Let's hear more about it.

6. Am enclosing an old article from MICRO that has information I found useful. A most useful column would be abstracts (and copies) of articles from the many magazines available. I get Creative Computing and would be glad to share comments / abstracts on that with another subscriber, i.e. alternate months or each take 1/2 (or 1/3) with 3 people each month.

Now for some ideas and questions generated by the fact that your article made me stop and think about this organization that I joined in 'blind' ignorance looking for 'freebies' / 'cheapies' but also information.

- a. The newsletter is good. Don't see where I can aid in assembly but will certainly try and contribute.
- b. What is NOVAPPLE? What is the relationship with 'Pi'?
- c. Does any member have a program that will

'pick up' the catalog from all my disks, collect it as data, allow me to add notes, and then sort

continued

it in disk no. and/or alpha sequence?

d. How about a comparison of the stores in the area on items such as books, programs, knowledge, helpfulness, variety of stock, price differences, etc. I have just sent out 7 RFPs for printer, interface and service contract. Will be glad to share the results when received.

e. Would like to see classes on selected subjects such as both interest groups and also like 'How do I combine hi-res with text?'

f. Any tutors / helpers in the group? For things like 'e.' and also aid on my ROMPLUS and why Applesoft Hi-Res behaves peculiarly (mine is in RAM).

g. An exchange / classified col. I would like an economical Applesoft ROM. Would like to sell / trade my like-new carrying case for \$15-20. Have built a box instead.

h. How about a service contract thru the club for APPLE owners?

i. How about, or do we have, a good demo disk to just turn on and let run for high school shows, etc.?

Sandy, this got verbose but it is your fault. You got me thinking and that is always danterous."

(Editor's note: (Many of the above suggestions already exist, or are in the process of being established.

1. We do have a software review section in the newsletter, though it is not in columnar form.

2. There is now a catalog list for the library disks.

3. SIG groups are needed and have been discussed. We need volunteers to start them off.

4. A classified ads column does exist in our newsletter.

Other suggestions are good and should be considered. Thank you, Charles, for taking the time to write this thoughtful letter.)

A QUICK AND DIRTY RAM TEST

by Jim Kelly

Here is a short program to check out your APPLE's read-write memory. I wrote it to test a set of EL-CHEAPO memory expansion chips. It's easy to enter since it takes less than 80 bytes. It runs very fast if there are no errors. That's the quick part. The dirty part is that it probably doesn't test everything that can go wrong. Specifically, since it stores and then loads immediately, it doesn't check the ability to hold a charge for a full refresh cycle. Anyway, it will find most other faults such as dead cells or bad connections.

To use the program, first enter from the monitor the program shown in the listing. It can of course be saved with a "300.34DW" command. Next the pointers must be loaded in memory locations \$00 through \$03 as follows:

```
AT:      ENTER:
$00      $00
$01      Highest page to be tested
$02      $00
$03      Lowest page to be tested
```

For example, if you were testing the second 16K of RAM, \$4000 through \$7FFF, enter the following:

```
*0:00 7FF 00 40
```

Note that the program won't check a portion of a page. Also, since the program resides in page 3, it can't check any lower than page 4.

After entering the pointers, simply enter "300G". If all is well, the cursor will return after a very short wait and nothing else will happen. If the program finds an error, your APPLE will beep and print an error message.

To understand the error message you must know how the program tests RAM. It stores and then loads from each location a bit pattern. Then it checks that it loaded what it stored. Two bit patterns are used. First it uses 10101010 which is \$AA. Then it uses 01010101 which is \$55. If an error is found, the error message printed consists of two 4-digit hex numbers. The first two digits of the first number are the pattern stored in memory. The second two digits are the pattern returned from memory. The second four digit number gives the address at which the error was found. For example:

```
5557 6F4B
```

indicates a error at address \$6F4B. To find out which bit was faulty, first convert the first two digits to binary. Then convert the second two to binary and write them under the first pattern. Any bit in the second line which does not match the top line is faulty. In the example, it would look like this:

```
01010101
01010111
      X
```

So the second bit from the right has the problem. The address indicates the fault is in the second 16K of memory. If you are using 16K RAM chips, the bad chips would be second from the left in the second row of RAM chips from the front. (The chips are in opposite order from the bits they represent.) If you have a problem, it's a good idea to swap the chip with a good chip (turn off the APPLE first!). Then reload and rerun the program to see if the problem follows the chip. If not, there is probably a problem in the APPLE itself.

If a whole chip gives a faulty load, or if you have accidentally included some ROM or vacant memory, the computer will beep like mad as it frantically prints out error messages. The only way to stop the program is to hit RESET. If pages 4 through 7 are included in your test, expect to see funny things on your screen. This is just the result of leaving \$55 on the page 1 text locations.

The program makes use of Monitor routines but seems to run OK on both the "old" Monitor and the AUTOSTART Monitor.

QUICK AND DIRTY RAM TEST - Listing

```
300: A0 00 E6 01 C6 01 88 A9
308: AA 85 04 91 00 B1 00 C5
310: 04 F0 03 20 2F 03 A9 55
318: 85 04 91 00 B1 00 C5 04
320: F0 03 20 2F 03 8 00 DE
328: A5 01 C5 03 D0 D6 60 AA
330: 98 48 A5 04 A8 20 40 F9
338: A9 A0 20 F0 FB 20 3A FF
340: 68 AA A5 01 A8 20 40 F9
348: 20 62 FC 8A A8 60
```

ERRATA:

The following program was omitted from the Librarian's review of Volume 1 (Utilities):

MEMORY SPY - If it's in memory this hex/ASCII dump program will help you find it! (Our thanks to H. Richoux for this one.)